MOTOROLA
Semiconductor Products Inc.

AN-864
Application Note

INTERFACING MULTIPLEXED BUS PERIPHERALS WITH NON-MULTIPLEXED MPUs

INTRODUCTION

In general, 8-bit microprocessor buses may be divided into two major groups: multiplexed and non-multiplexed. A microprocessor (MPU) which has a non-multiplexed bus configuration requires two separate buses: an address bus and a data bus. However, an MPU which has a multiplexed bus configuration shares one bus for both address and data; i.e., the lower eight bits of address are output on the same lines that are used for the data bus. To accomplish this, a multiplexed bus cycle must be divided in two parts. During the first part of the cycle, the lower eight bits of address are output on the multiplexed bus pins. During the second part of the cycle, data is input or output via the multiplexed bus pins.

It is important that the two parts of the cycle be identified by both the multiplexed and non-multiplexed MPUs. For this reason, an address strobe line (AS or ALE) is used to indicate when the address is valid. During the second part of the cycle, the multiplexed bus lines are used to transfer data, and a data strobe line (DS) is used to indicate this.

Interfacing a multiplexed bus MPU (such as MC6803 or MC146805E2) with non-multiplexed bus peripherals is relatively straightforward. The interface can be accomplished by simply providing an external latch to capture the multiplexed address for the entire bus cycle. However, providing an interface between a non-multiplexed bus MPU (such as the MC6800 and the MC6809) and multiplexed bus peripherals is more difficult.

Generating the required address and data strobes, while also guaranteeing that the address and data are on and off the multiplexed bus at the required times, is not possible with most non-multiplexed MPUs. Methods for providing an interface between non-multiplexed bus MPUs and multiplexed bus peripherals are discussed below.

SUGGESTED INTERFACING METHODS

One method that is often used, but is not too efficient, employs the I/O lines of a peripheral interface adapter (PIA) to provide the multiplexed bus signals to the peripherals. This method, while fine for single-chip computers, requires excessive software, I/O lines, and execution time.

Another method, for allowing a non-multiplexed bus MPU to read or write multiplexed bus peripherals, is to consider the multiplexed bus as a two-register device. One register could be the address latch and the other could be the data register. By connecting the non-multiplexed bus to the multiplexed bus, as shown in Figure 1, the address strobe (AS) is generated by any MPU write to an even address (A0 is low). Although AS causes the state of the multiplexed pins to be captured in the address latch, no data is transferred until the data strobe (DS, $\overline{RD}$, or $\overline{WR}$) is generated. The data strobe is not generated until a specific decoded, odd address occurs; i.e., A0 is high. Thus, a valid access can only be made by first writing to the location ($00-$3F) of the byte to be accessed at an even address, and then by either reading from or writing to the decoded odd address. The software subroutine required to generate the address strobe (AS) and data strobe (DS) signals for the interface shown in Figure 1 is listed in Figure 2. The location to be accessed is contained in accumulator A and data is transferred via accumulator B. The following example clears register A (location $0A) and then reads the data at location $25.

```
LDAA    #$0A     Load $0A into accumulator A for
                 location
CLRB             Load $00 into accumulator B for
                 data
BSR     WRITE    Call subroutine to write data
CLRA             Load $00 into accumulator A for
                 location
BSR     READ     Call subroutine to read data
STAB    DATA     Data is returned to accumulator B
                 and saved
```

The schematic diagram shown in Figure 1 provides an interface between Motorola devices: MC6800/MC6802/MC6809 MPU interfaced with an MC146818 Real Time Clock or MC146823 CMOS Parallel Interface. Since both the MC146818 and the MC146823 are implemented with the Motorola MOTEL circuit (for MOTorola and IntEL bus compatibility), they can be interfaced with either of the two most common multiplexed bus structures. The hardware required for interfacing a competitor's non-multiplexed MPU with either the MC146818 or MC146823 is shown in Figure 3.

Both the MC146818 and MC146823 are CMOS devices and may be used with a battery back-up; however, power-down is different for each device. Also, on both devices $\overline{CE}$ is used to isolate the device from the bus except as follows: (1) when AS is high on the MC146823 and (2) on the MC146818 $\overline{CE}$ must be valid before the falling edge of AS and it must remain valid until the trailing edge of DS. For these reasons, the CMOS inverter shown in Figures 1 and 3, is used to provide $\overline{CE}$ from the +5 V system power source. With the inverter connected, $\overline{CE}$ always remains enabled as long as system power is present. With $\overline{CE}$ enabled, accesses are controlled by gating the address decoding into AS and DS signals. When system power is removed, the CMOS inverter disables $\overline{CE}$ and prevents any further MPU accesses.

NOTE

If a battery back-up for the MC146818 or MC146823 is not used, the $\overline{CE}$ pin can be grounded.

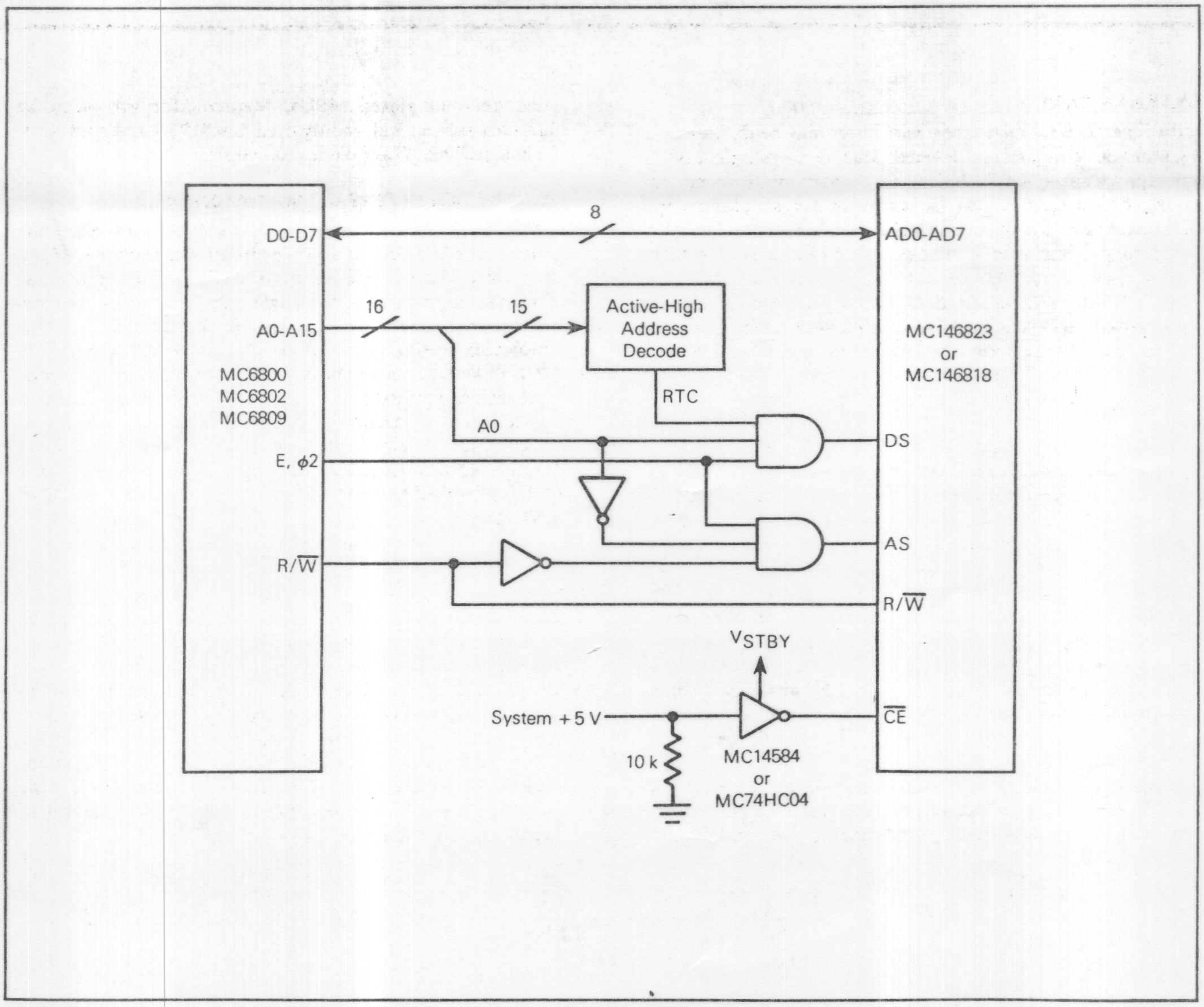

FIGURE 1 — MC146818/MC146823 and Motorola MPU
Non-Multiplexed Bus Interface Schematic Diagram

```
READ    STAA   RTC      Generate AS and latch data from
                        ACCA
        LDAB   RTC+1    Generate DS and load ACCB
                        with data read
        RTS             Return

WRITE   STAA   RTC      Generate AS and latch data from
                        ACCA
        STAB   RTC+1    Generate DS and store data from
                        ACCB
        RTS             Return
```

FIGURE 2 — Subroutine for Controlling Interface Between Non-Multiplexed Bus and MC146818/MC146823

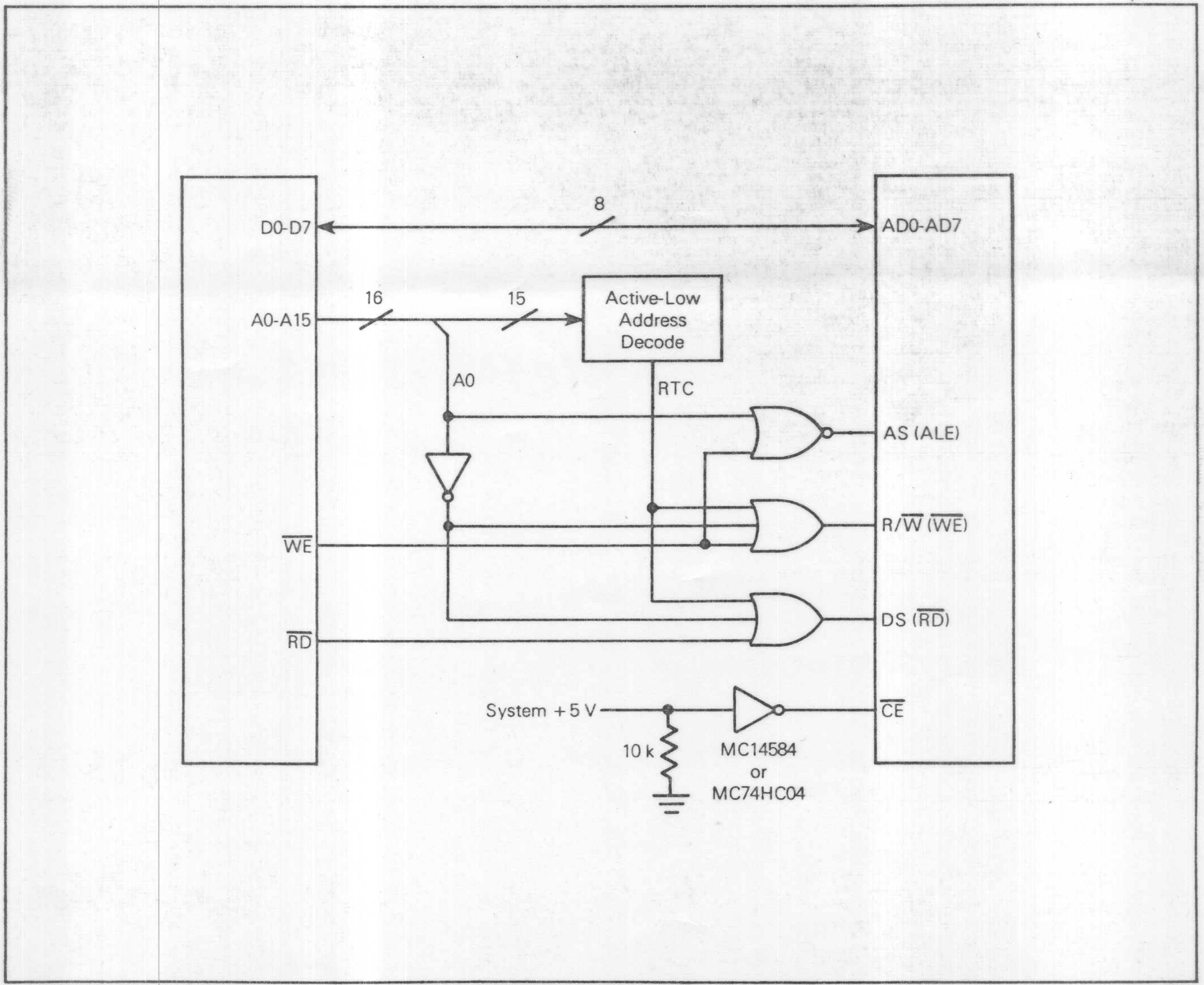

FIGURE 3 — MC146818/MC146823 and Competitor MPU Non-Multiplexed Bus Interface Schematic Diagram